



P2M2HBVL/P2H IO-LINK
FUNCTION BLOCK FOR SIEMENS
PLC S7-1200/1500 WITH
BALLUFF BNI005H PROFINET IO-
LINK MASTER
QUICK START GUIDE

CONTENTS

PREFACE	3
1. Setting P2M\P2H Slave with BNI005H Balluff IO-Link Master	4
2. Importing function block library	5
3. Process Data Function Block Instruction.....	7
3.1. Add instance of function block	7
3.2. Using the instruction.....	9
3.2.1. Solenoid Valves Control (Cyclic).....	9
3.2.2. Monitoring Status Bits (Cyclic)	10
4. Parameter Data Example Code	11
4.2. Add instance of function block	11
4.3. Using the instruction.....	13
4.3.1. Reading parameters.....	13
4.3.2. Writing Parameters.....	15
APPENDIX.....	18

PREFACE

This Quick Start Guide (QSG) is designed to help integrate Parker Hannifin's P2M or P2H IO-link valve manifold into a Siemens PLC environment using Balluff BNI005H profinet IO-Link Master module. The QSG assumes that you are already using the balluff network interface profinet and that it is connecting and configured to the siemens PLC S7-1200/1500 via a profinet network in Tia portal environment.

The QSG is agnostic to IO Link Device Classification, such that it shall function the same whether you are controlling an A-Class or B-Class P2M / P2H Module. The guide will walk the user through obtaining the necessary files, importing/configuring the FB's, and initiating parameter reads and writes from/to the P2M / P2H IO-Link device.

The **"P2M_IOL_BNI005H_PD"** function block facilitates communication and handling of process data between PLC and the IO-Link slave device.

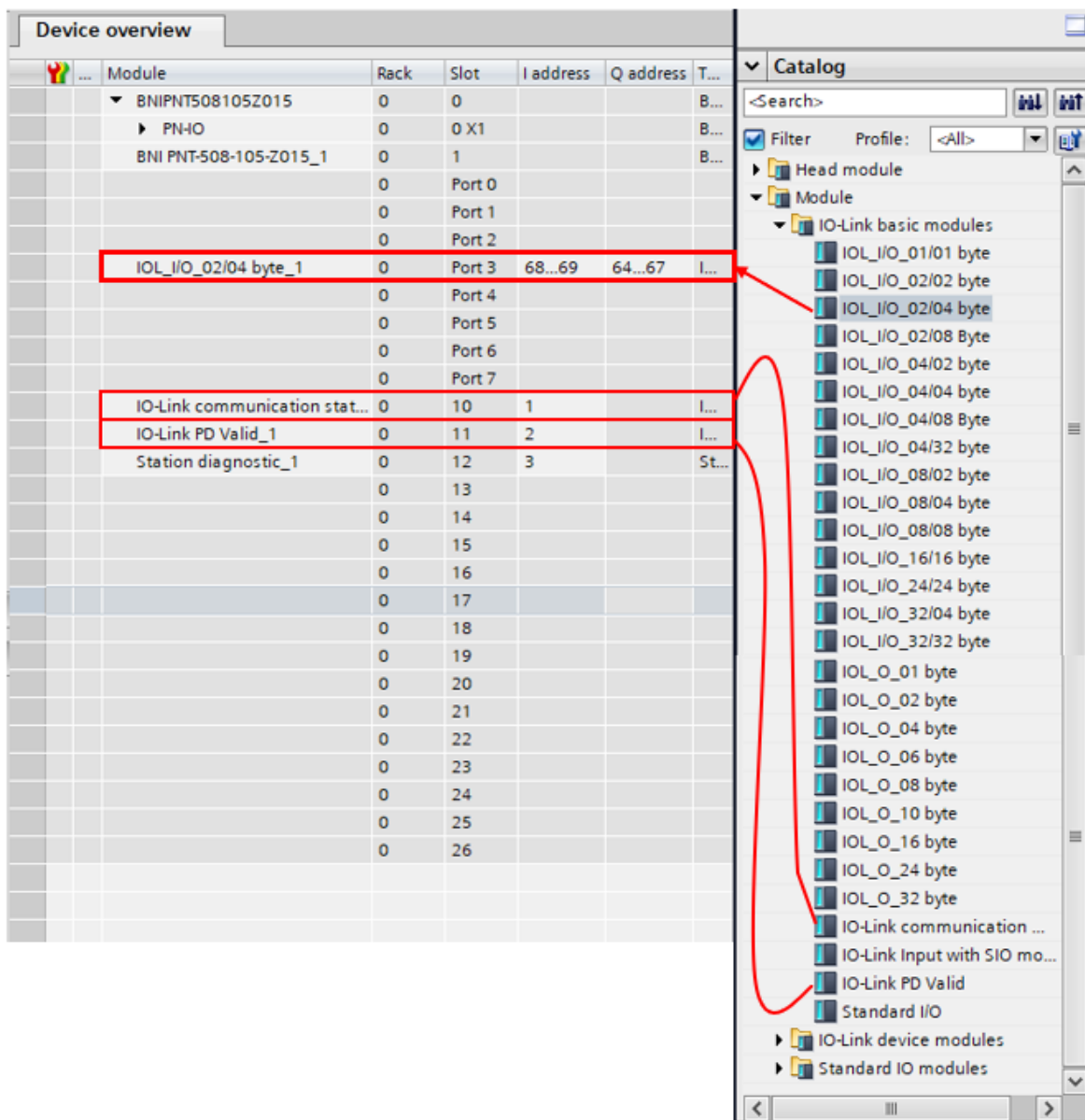
The **"P2M_IOL_BNI005H_PRM"** function block facilitates the call-up of the acyclic service data.

You can download resources such as the IODD configuration file, this QSG, a sample TIA portal file **"P2M2HBVL_P2HL_TIA_S71200_BNI005H_R0"** and the full P2M manual here:

<http://www.parker.com/pdn/io-Link>

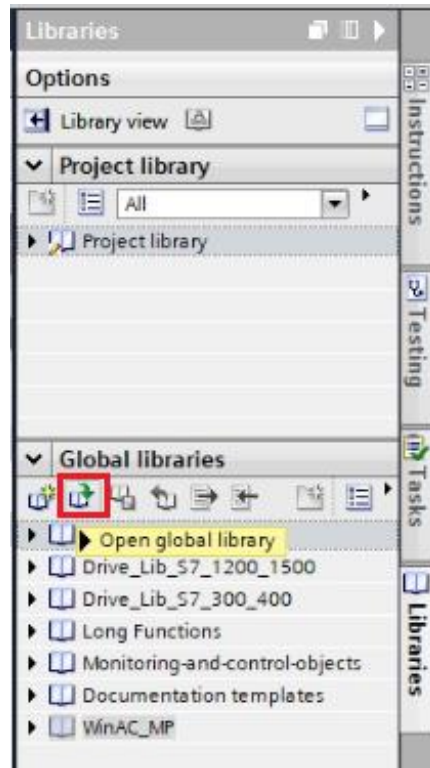
1. SETTING P2M\P2H SLAVE WITH BNI005H BALLUFF IO-LINK MASTER

In the project tree double click “Devices & networks” move to “Network view” double click the Balluff IO-Link master, drag and drop “**IOL_I/O_02/04 byte**”, **IO-Link Communication state** and **IO-Link PD Valid** from Hardware catalog >> Module>>IO-Link basic modules to “Device view”.

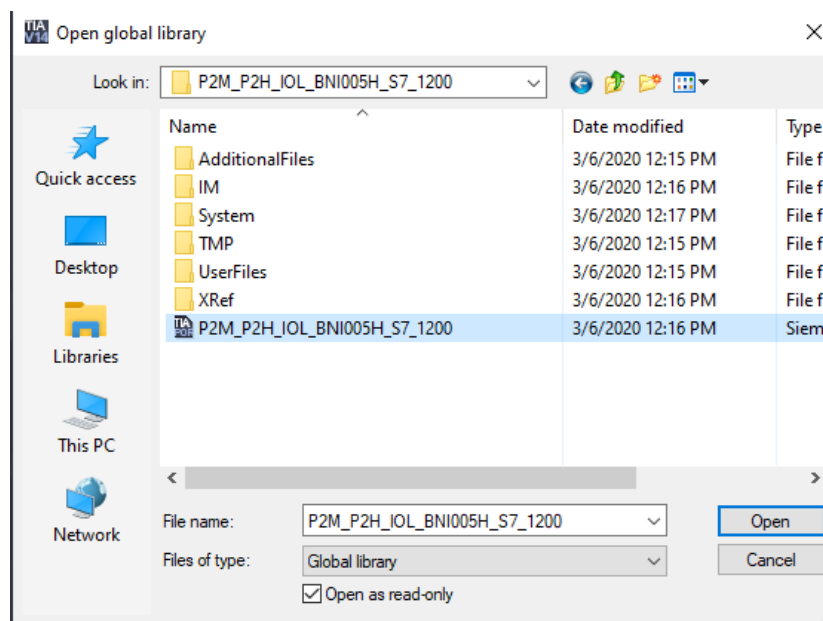


2. IMPORTING FUNCTION BLOCK LIBRARY

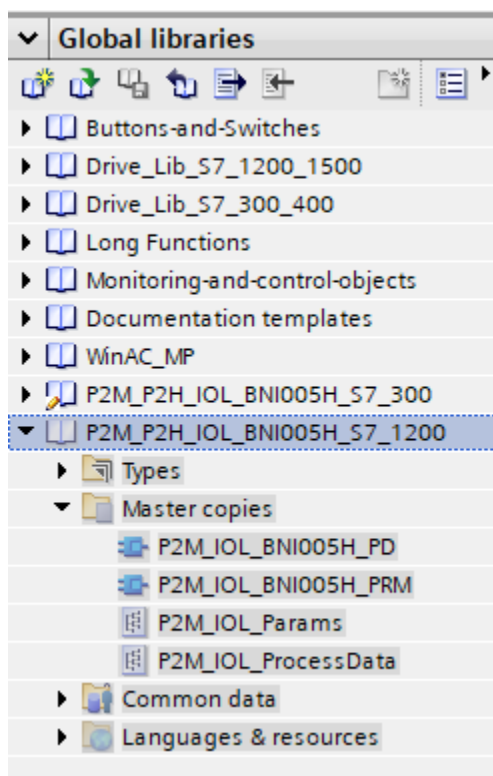
In Tia portal environment at right side, open the tab “Global Libraries” by clicking on “Libraries”, then select “open global library”.



In “Open global Library” view move to your stored library function block folder “P2M2HBVL_P2HL_TIA_S71200_BNI005H_R0” and select “.al14” file then click **open**

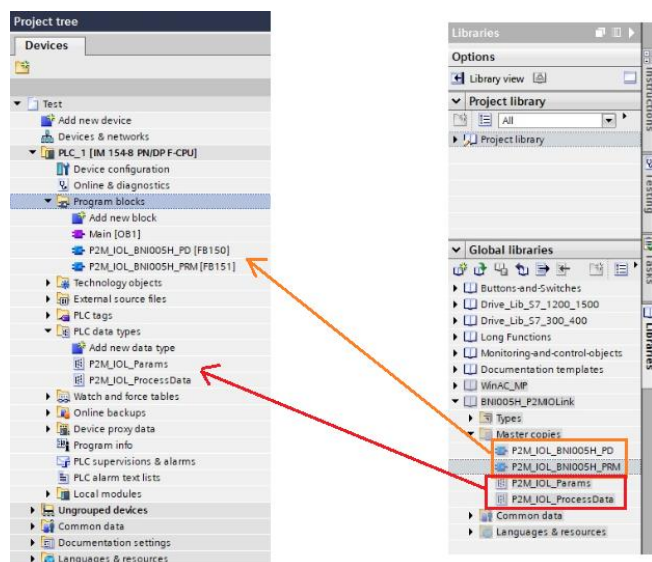


You should then see the new Library “P2M_P2H_IOL BNI005H_S7_1200” in the Global Libraries list, click on your imported Library title to view your function block from drop-down menu.



Drag and drop type data “P2M_IOL_Params” and the “P2M_IOL_Params” from Global libraries to the project tree “PLC data types” folder.

Then drag and drop function block “P2M_IOL_BNI005H_PD” and “P2M_IOL_BNI005H_PRM” to the project tree folder “Program blocks”.

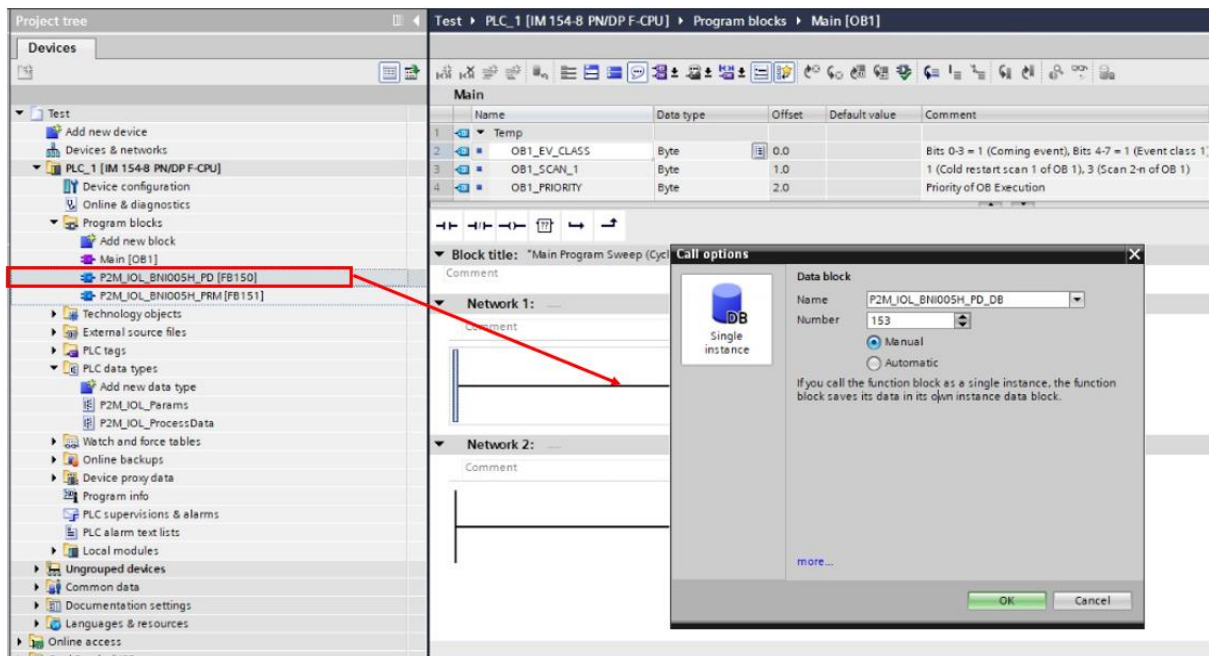


3. PROCESS DATA FUNCTION BLOCK INSTRUCTION

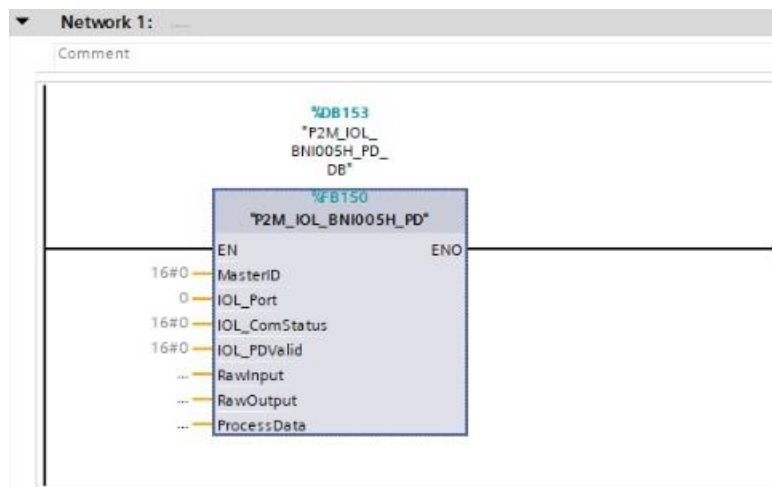
The “P2M_IOL_BNI005H_PD” function block simplifies the usage of Parker P2M and P2H IO-link devices with Siemens S7-1200/1500 PLCs when connected via profinet to Balluff BNI005H IO-Link Master. Data is mapped to user-friendly control and diagnostic tags on the PLC side.

3.1. Add instance of function block

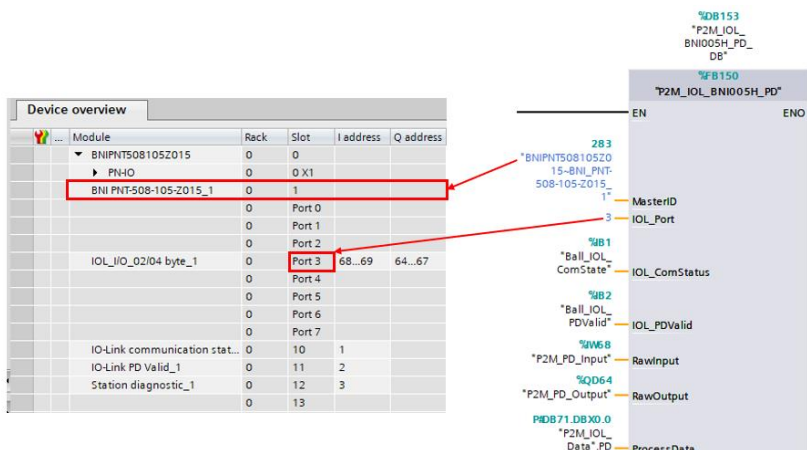
Add instance of instruction to an empty Network of ladder, drag and drop the function block “P2M_IOL_BNI005H_PD” onto the selected Network, create data instance associated with function block then click Ok.



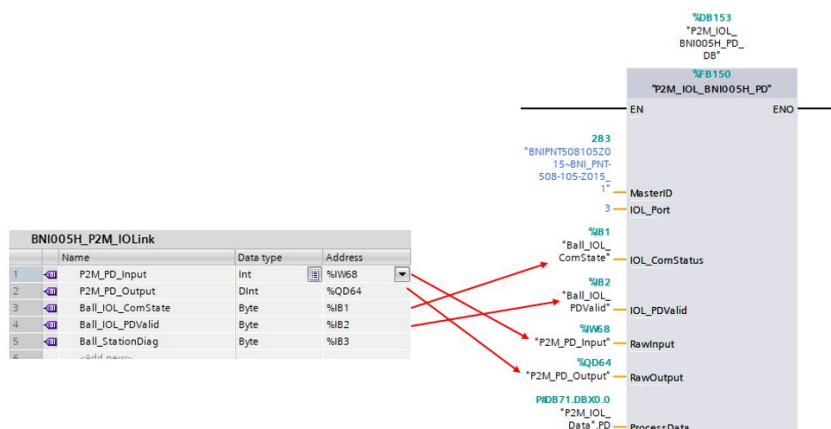
function block view with default data in the Network



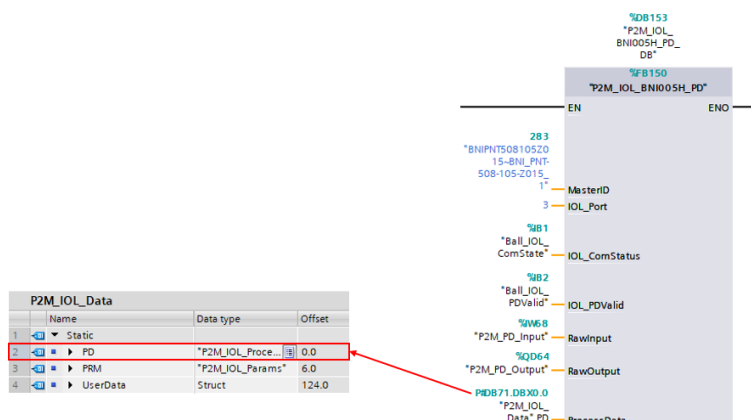
Read **MasterID** and **IOL_Port** from “Devices & networks” > “Device Overview”. The **IOL_Port** is **dependent on the physical port the P2M / P2H is connected to!** If you choose the incorrect port the function block will not work, and undesired results are likely. IO-Link is sensitive to the port which it has been assigned to communicate on.



Point IOL_ComStatus, IOL_PDValid, RawInput and RawOutput fields to the “PLC Tags” associated with IO-Link Basic module that the P2M / P2H is connected to.



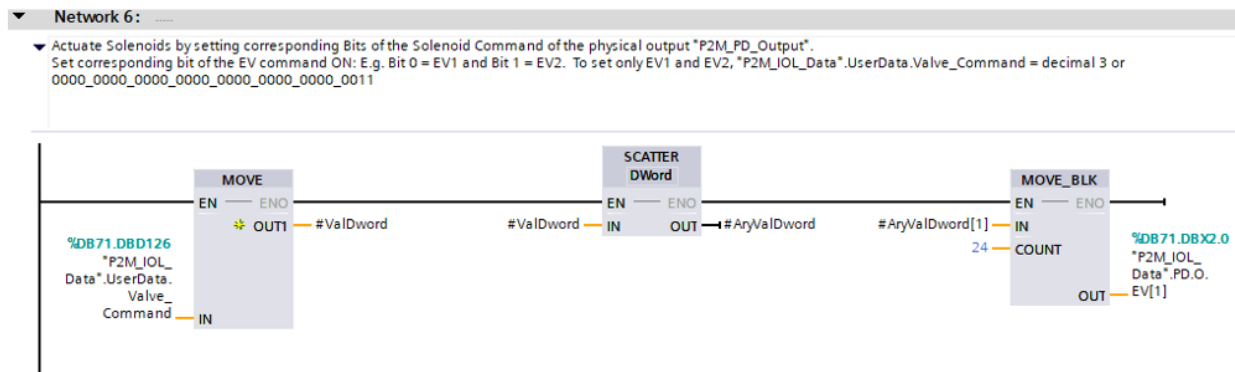
Point “ProcessData” to a variable type “P2M_IOL_ProcessData” created in a user data block



3.2. Using the instruction

It is important to note the difference between cyclic and acyclic data. Process Data (cyclic) is updated without a request; whereas Parameter Data (acyclic) requires the program to toggle a bit to pull data. Cyclic data includes input module status, and valve output control. This means that created variable “P2M_IOL_Data”.PD.I.xxx and “P2M_IOL_Data”.PD.O.EV[xx] are live tags with real data just by calling the Function Block. See appendix for all data points available. See ladder logic examples below:

3.2.1. Solenoid Valves Control (Cyclic)

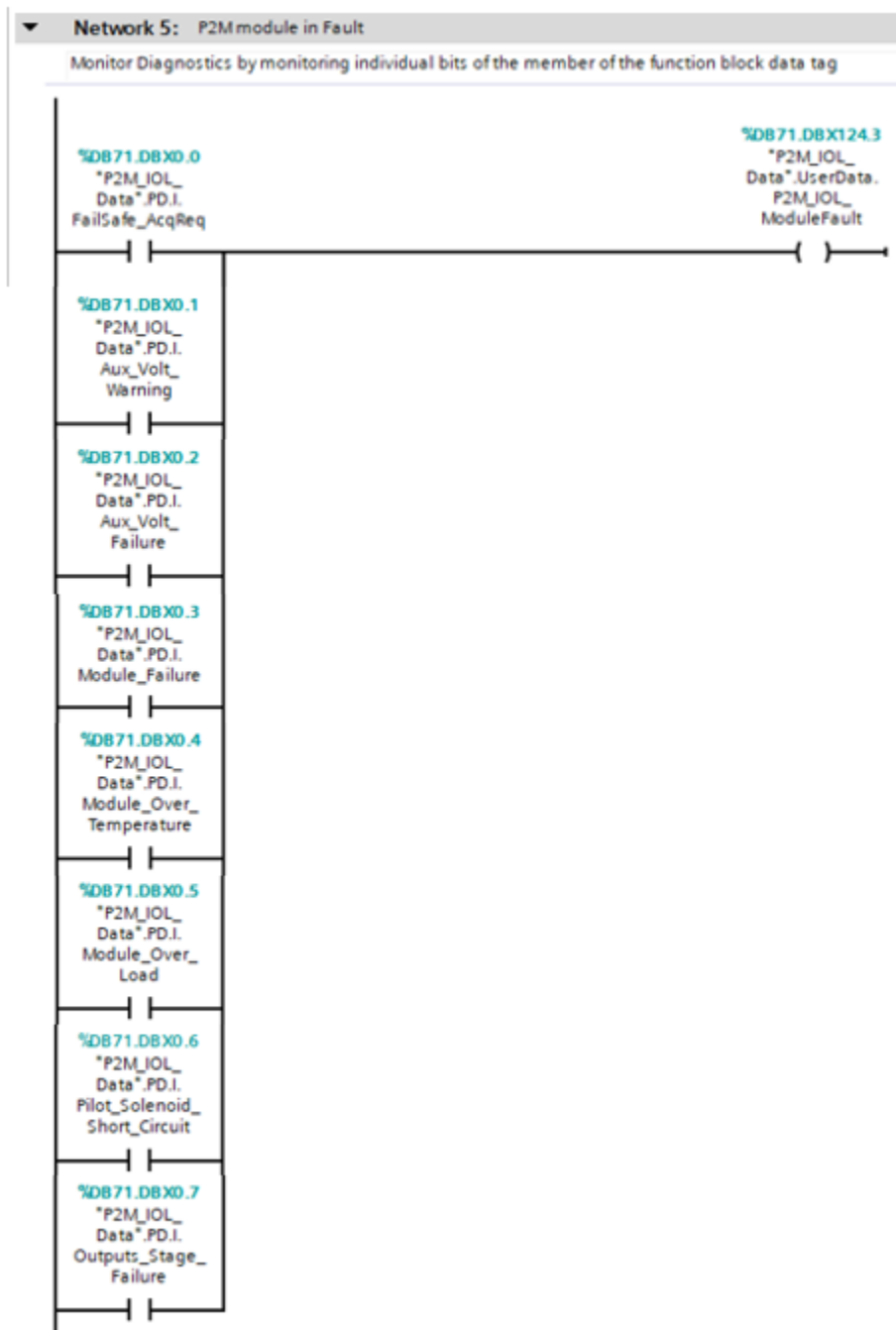


Example command

The “P2M_IOL_Data”.UserData.Valve_Command value is a decimal number indicating which valves you want Switching ON .

Value correlation to solenoids address	
“P2M_IOL_Data”.UserData.Valve_Command = 1	“P2M_IOL_Data”.PD.O.EV[1]=1
“P2M_IOL_Data”.UserData.Valve_Command = 2	“P2M_IOL_Data”.PD.O.EV[2]=1
“P2M_IOL_Data”.UserData.Valve_Command = 4	“P2M_IOL_Data”.PD.O.EV[3]=1
“P2M_IOL_Data”.UserData.Valve_Command = 8	“P2M_IOL_Data”.PD.O.EV[4]=1
“P2M_IOL_Data”.UserData.Valve_Command = 16	“P2M_IOL_Data”.PD.O.EV[5]=1
.	
.	
.	
“P2M_IOL_Data”.UserData.Valve_Command = 8 388 608	“P2M_IOL_Data”.PD.O.EV[24]=1

3.2.2. Monitoring Status Bits (Cyclic)



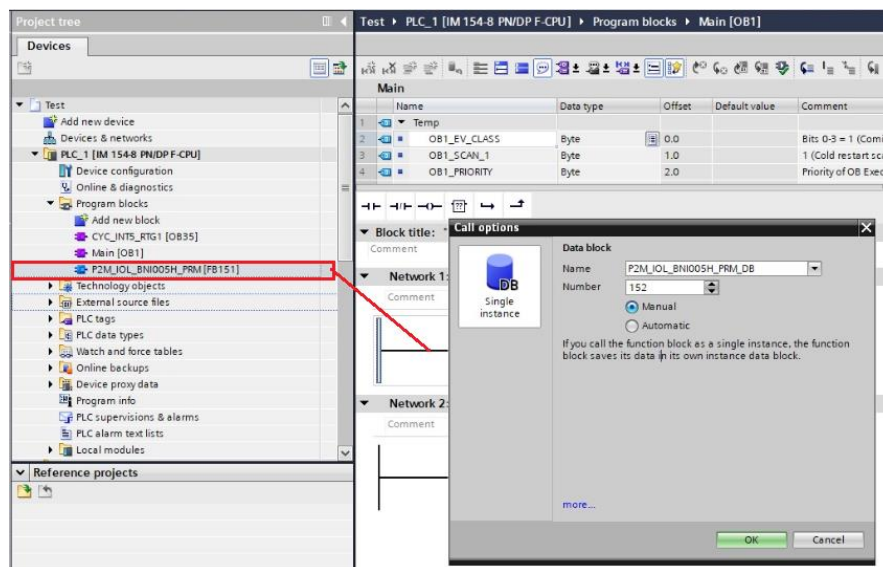
4. PARAMETER DATA EXAMPLE CODE

Note:** It is important to not initiate multiple read or write requests at the same time. Write your logic such that only one of the request bits is turned on at a time and wait for the done bit to go high before executing the next request.

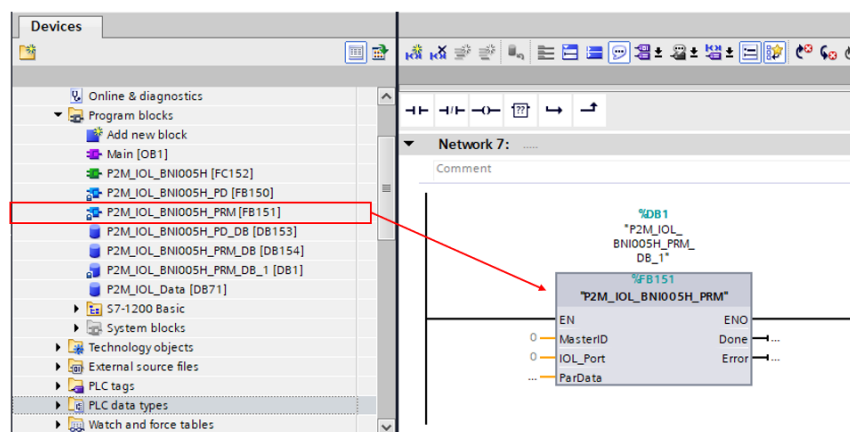
The “P2M_IOL_BNI005H_PRM” function block simplifies the usage of Parker P2M and P2H IO-link devices with Siemens S7-1200/1500 PLCs when connected via profinet to Balluff BNI005H IO-Link Master. The Function block facilitates the read/write acyclic data between the PLC and the Parker P2M or P2H IO-Link device. We assume that the Global Libraries importing steps are already done by following the described steps above.

4.2. Add instance of function block

From “Program Blocks” Drag and drop the function block “P2M_IOL_BNI005H_PRM” to add an instance of instruction to an empty network of ladder

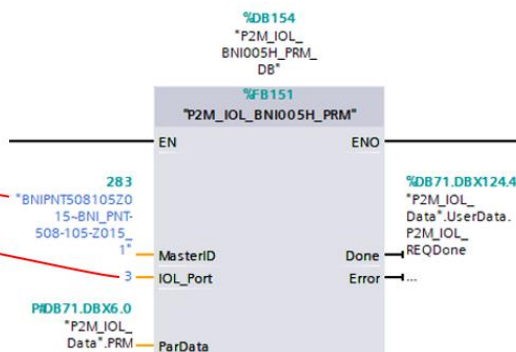


Assign a data block to the instance, define the name and the number. View the create instance below with default data.



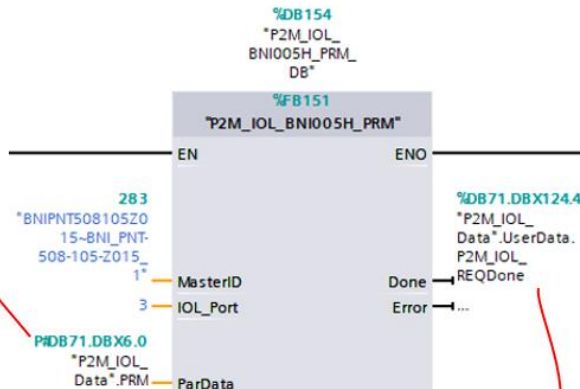
Create Tags necessary for operation. Read and set **MasterID** and **IOL_Port** from “Devices & networks” > “Device Overview”. The **IOL_Port** is dependent on the physical port the P2M / P2H is connected to! If you choose the incorrect port the function block will not work, and undesired results are likely. IO-Link is sensitive to the port which it has been assigned to communicate on.

Device overview					
Module	Rack	Slot	I address	Q address	
BNIPNT508105Z015	0	0			
PN-IO	0	0 X1			
BNI PNT-508-105-Z015_1	0	1			
	0	Port 0			
	0	Port 1			
	0	Port 2			
IOL_I/O_02/04 byte_1	0	Port 3	68...69	64...67	
	0	Port 4			
	0	Port 5			
	0	Port 6			
	0	Port 7			
IO-Link communication stat...	0	10	1		
IO-Link PD Valid_1	0	11	2		
Station diagnostic_1	0	12	3		
	0	13			



Point “ParData” to a variable type “**P2M_IOL_Params**” created in a user data block and “Done” to a variable type BOOL created in an user data block.

P2M_IOL_Data			
Name	Data type	Offset	
1 Static			
2 PD	*P2M_IOL_Process...	0.0	
3 PRM	*P2M_IOL_Params*	6.0	
4 UserData	Struct	124.0	
5 P2M_IOL_Connect...	Bool	124.0	
6 P2M_IOL_DeviceM...	Bool	124.1	
7 P2M_IOL_PDValid	Bool	124.2	
8 P2M_IOL_ModuleF...	Bool	124.3	
9 P2M_IOL_REQDone	Bool	124.4	
10 ToggleRd_param	Bool	124.5	
11 ToggleWr_AuxLevel	Bool	124.6	
12 ToggleWr_ClearCo...	Bool	124.7	
13 Toggle_SysCmd_A...	Bool	125.0	
14 Toggle_SysCmd_S...	Bool	125.1	
15 Toggle_RestoreFac...	Bool	125.2	
16 Valve_Command	DInt	126.0	
17 Wr_Value_ClearCo...	DInt	130.0	
18 Rd_Value_AuxVolt...	Int	134.0	
19 Rd_Value_AuxLoLi...	Int	136.0	
20 Rd_Value_AuxHiLi...	Int	138.0	
21 Rd_Value_Channel...	DInt	140.0	
22 Rd_Value_Switchin...	Array[1..24] of DInt	144.0	
23 Wr_Value_AuxLoLi...	Int	240.0	
24 Wr_Value_AuxHiLi...	Int	242.0	

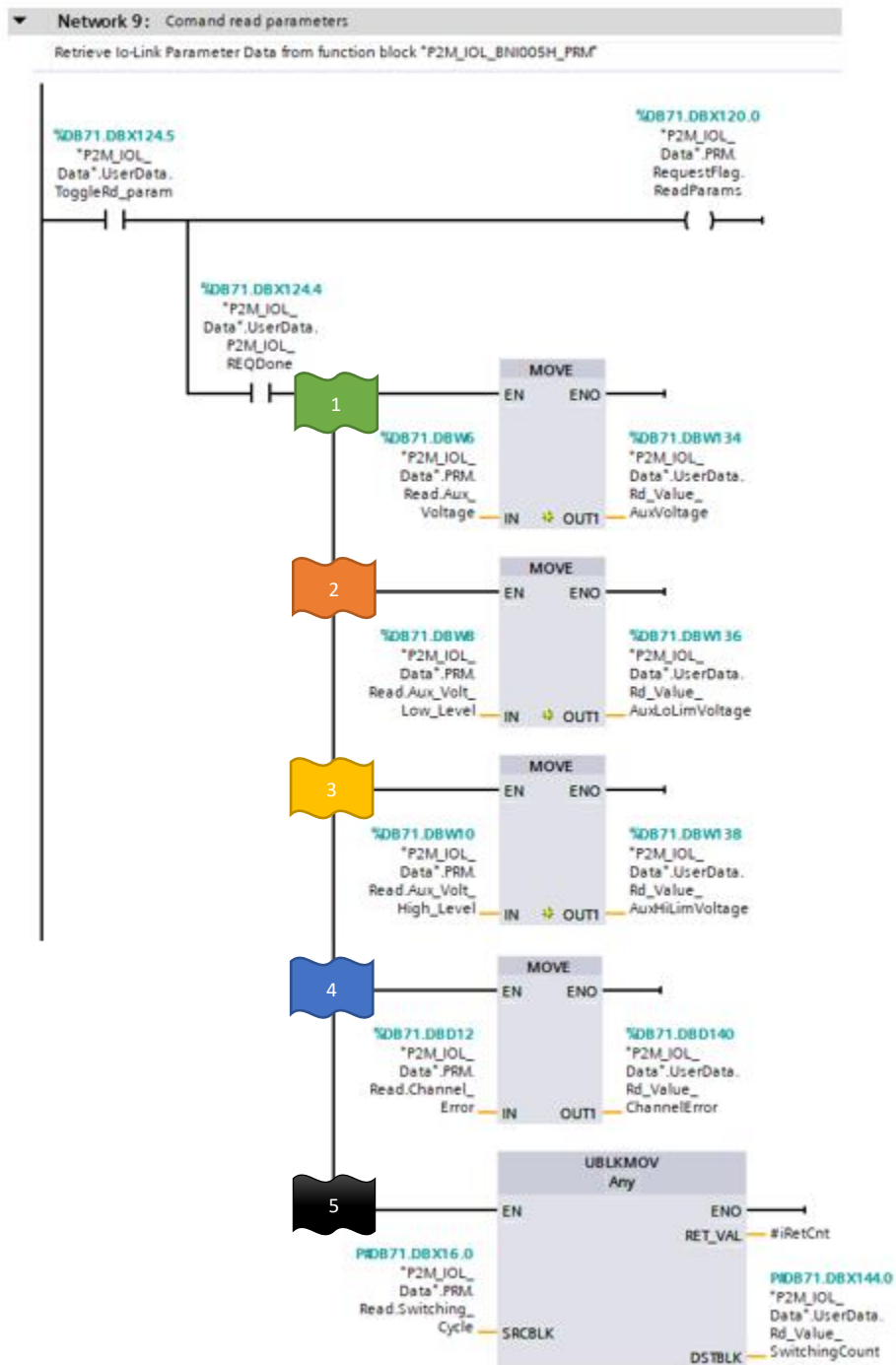


4.3. Using the instruction

The parameter data(acyclic) requires the program to toggle a bit to read or write data contained inside the slave device.

4.3.1. Reading parameters.

see below for logic example

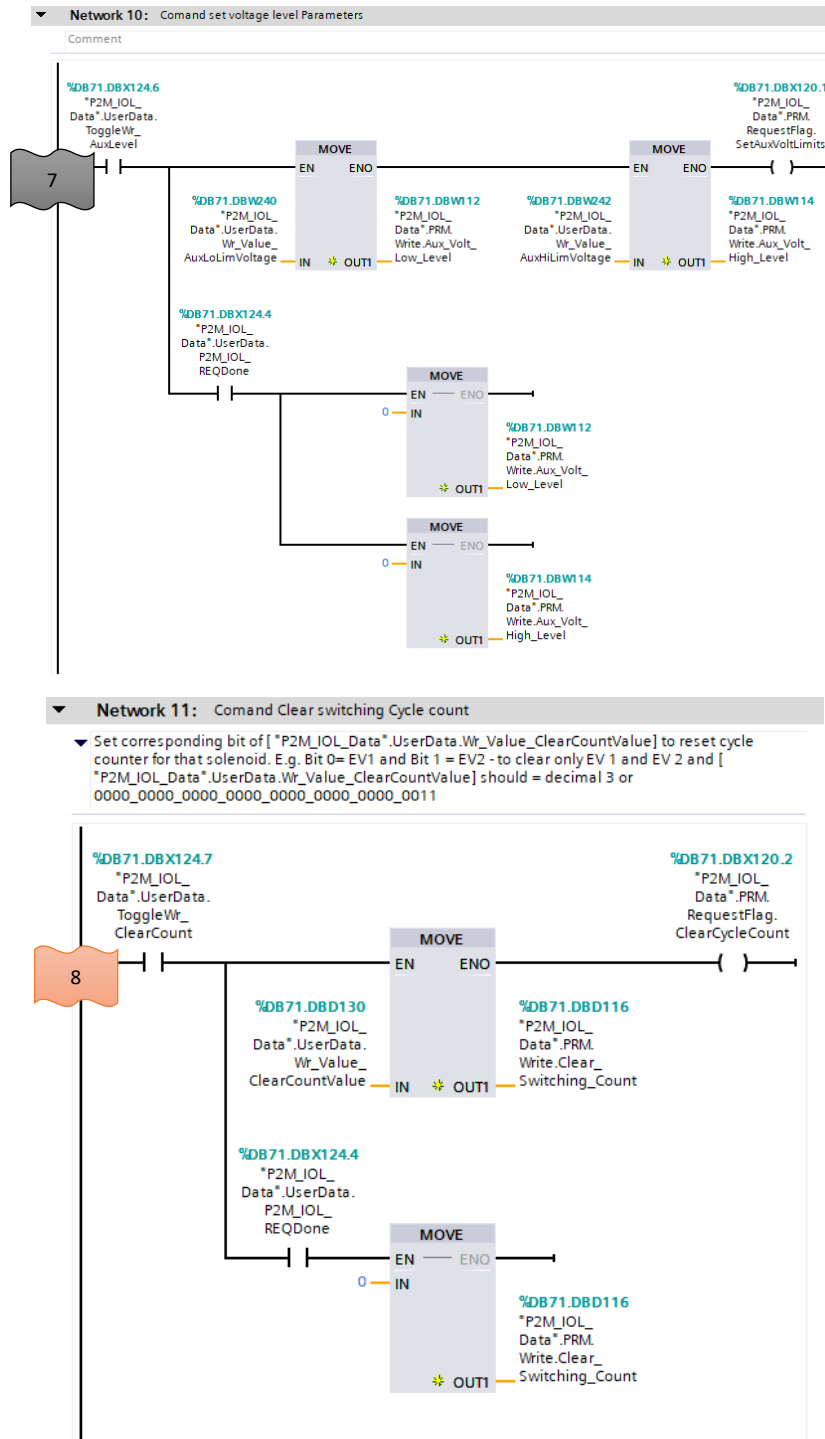


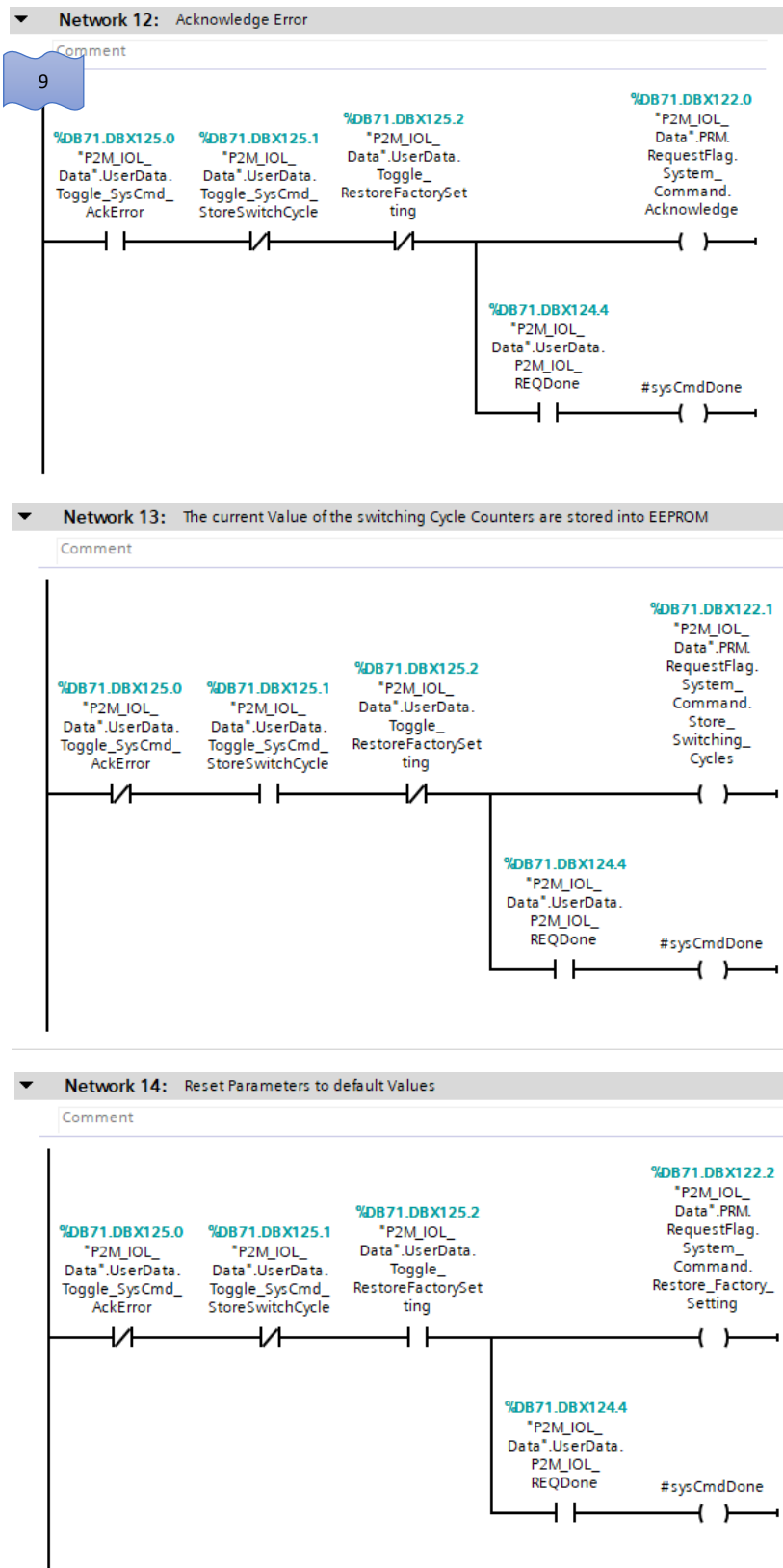
	Description	Index	Destination Element
1	Read Auxiliary Voltage	67	P2M_IOL_Data.PRM.Read.AuxVoltage
2	Read Auxiliary Voltage Low Limit Warning	70	P2M_IOL_Data.PRM.Aux_Volt_Low_Lev
3	Read Auxiliary Voltage High Limit Warning		P2M_IOL_Data.PRM.Aux_Volt_High_Lev
4	Read Channel Errors	66	P2M_IOL_Data.PRM.Channel_Error
5	Read Cycle Counters	64	P2M_IOL_Data.PRM.Switching_Cycle

4.3.2. Writing Parameters.

Writing Auxiliary Voltage Low and High Limits, clearing cycle counters example, send systems command.

Note: **When sending new limit setpoints to the P2M module, the values will not be written unless the high limit is more than one volt greater than the low limit, and greater than zero.**

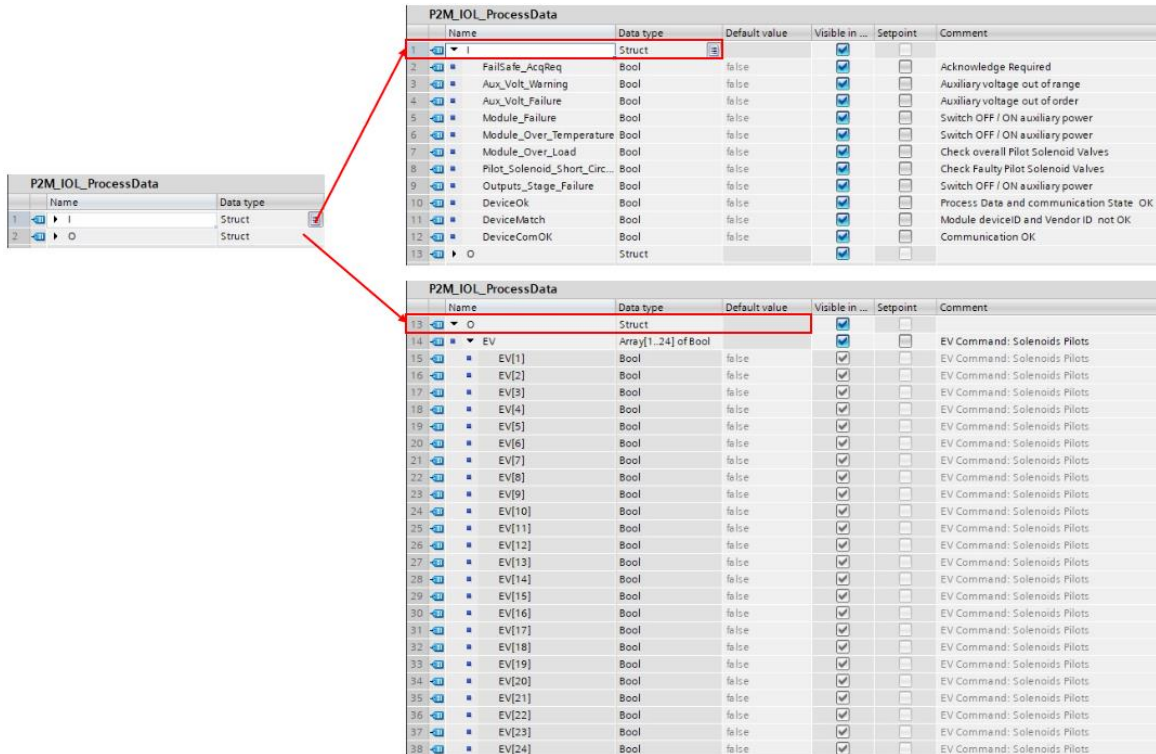




	Description	Index	Source Element		Source Length (bytes)
7	Write Auxiliary Voltage Low Limit Warning	70	P2M_IOL_Data.PRM.Write.Aux_Volt_Low_Level		4
8	Write Auxiliary Voltage High Limit Warning		P2M_IOL_Data.PRM.Write.Aux_Volt_High_Level		Note**: first 16bits=Low level, second 16bits=high level
9	Clear Cycle Counters	65	P2M_IOL_Data.PRM.Write.Clear_Switching_Count		4
	System Command	2	P2M_IOL_Data.PRM.RequestFlag.System_Command: Note: **By sending a system command request these values are written		1
			Command Value	Command Send	
			160	Acknowledge	
			161	Store Switching Cycles	
			130	Restore Factory Value	

APPENDIX

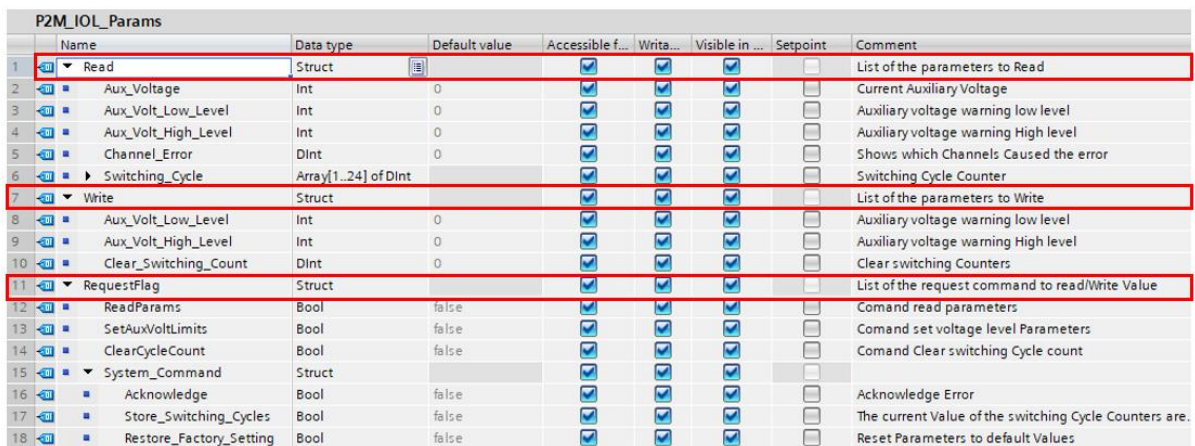
➤ **Process Data Structures:** User Defined Data Structures utilized by FB “P2M_IOL_BNI005H_PD”
From project tree\PLC click on “PLC data types” then double click to open “P2M_IOL_ProcessData” data types.



Name	Data type	Default value	Visible in ...	Setpoint	Comment
1	Struct		<input checked="" type="checkbox"/>	<input type="checkbox"/>	
2	FailSafe_AcqReq	Bool	false	<input checked="" type="checkbox"/>	Acknowledge Required
3	Aux_Volt_Warning	Bool	false	<input checked="" type="checkbox"/>	Auxiliary voltage out of range
4	Aux_Volt_Failure	Bool	false	<input checked="" type="checkbox"/>	Auxiliary voltage out of order
5	Module_Failure	Bool	false	<input checked="" type="checkbox"/>	Switch OFF / ON auxiliary power
6	Module_Over_Temperature	Bool	false	<input checked="" type="checkbox"/>	Switch OFF / ON auxiliary power
7	Module_Over_Load	Bool	false	<input checked="" type="checkbox"/>	Check overall Pilot Solenoid Valves
8	Pilot_Solenoid_Short_Circ...	Bool	false	<input checked="" type="checkbox"/>	Check Faulty Pilot Solenoid Valves
9	Outputs_Stage_Failure	Bool	false	<input checked="" type="checkbox"/>	Switch OFF / ON auxiliary power
10	DeviceOk	Bool	false	<input checked="" type="checkbox"/>	Process Data and communication State OK
11	DeviceMatch	Bool	false	<input checked="" type="checkbox"/>	Module deviceID and Vendor ID not OK
12	DeviceComOK	Bool	false	<input checked="" type="checkbox"/>	Communication OK
13	Struct		<input checked="" type="checkbox"/>	<input type="checkbox"/>	
14	EV	Array[1..24] of Bool	<input checked="" type="checkbox"/>	<input type="checkbox"/>	EV Command: Solenoids Pilots
15	EV[1]	Bool	false	<input checked="" type="checkbox"/>	EV Command: Solenoids Pilots
16	EV[2]	Bool	false	<input checked="" type="checkbox"/>	EV Command: Solenoids Pilots
17	EV[3]	Bool	false	<input checked="" type="checkbox"/>	EV Command: Solenoids Pilots
18	EV[4]	Bool	false	<input checked="" type="checkbox"/>	EV Command: Solenoids Pilots
19	EV[5]	Bool	false	<input checked="" type="checkbox"/>	EV Command: Solenoids Pilots
20	EV[6]	Bool	false	<input checked="" type="checkbox"/>	EV Command: Solenoids Pilots
21	EV[7]	Bool	false	<input checked="" type="checkbox"/>	EV Command: Solenoids Pilots
22	EV[8]	Bool	false	<input checked="" type="checkbox"/>	EV Command: Solenoids Pilots
23	EV[9]	Bool	false	<input checked="" type="checkbox"/>	EV Command: Solenoids Pilots
24	EV[10]	Bool	false	<input checked="" type="checkbox"/>	EV Command: Solenoids Pilots
25	EV[11]	Bool	false	<input checked="" type="checkbox"/>	EV Command: Solenoids Pilots
26	EV[12]	Bool	false	<input checked="" type="checkbox"/>	EV Command: Solenoids Pilots
27	EV[13]	Bool	false	<input checked="" type="checkbox"/>	EV Command: Solenoids Pilots
28	EV[14]	Bool	false	<input checked="" type="checkbox"/>	EV Command: Solenoids Pilots
29	EV[15]	Bool	false	<input checked="" type="checkbox"/>	EV Command: Solenoids Pilots
30	EV[16]	Bool	false	<input checked="" type="checkbox"/>	EV Command: Solenoids Pilots
31	EV[17]	Bool	false	<input checked="" type="checkbox"/>	EV Command: Solenoids Pilots
32	EV[18]	Bool	false	<input checked="" type="checkbox"/>	EV Command: Solenoids Pilots
33	EV[19]	Bool	false	<input checked="" type="checkbox"/>	EV Command: Solenoids Pilots
34	EV[20]	Bool	false	<input checked="" type="checkbox"/>	EV Command: Solenoids Pilots
35	EV[21]	Bool	false	<input checked="" type="checkbox"/>	EV Command: Solenoids Pilots
36	EV[22]	Bool	false	<input checked="" type="checkbox"/>	EV Command: Solenoids Pilots
37	EV[23]	Bool	false	<input checked="" type="checkbox"/>	EV Command: Solenoids Pilots
38	EV[24]	Bool	false	<input checked="" type="checkbox"/>	EV Command: Solenoids Pilots

➤ **Parameter Data Structures:** User Defined Data Structures utilized by FB
“P2M_IOL_BNI005H_PRM”

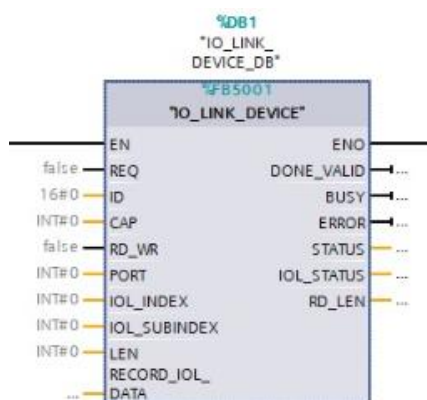
From project tree\PLC click on “PLC data types” then double click to open “P2M_IOL_Params” data types.



Name	Data type	Default value	Accessible f...	Writa...	Visible in ...	Setpoint	Comment
1	Read	Struct	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	List of the parameters to Read
2	Aux_Voltage	Int	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Current Auxiliary Voltage
3	Aux_Volt_Low_Level	Int	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Auxiliary voltage warning low level
4	Aux_Volt_High_Level	Int	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Auxiliary voltage warning High level
5	Channel_Error	DInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Shows which Channels Caused the error
6	Switching_Cycle	Array[1..24] of DInt		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Switching Cycle Counter
7	Write	Struct	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	List of the parameters to Write
8	Aux_Volt_Low_Level	Int	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Auxiliary voltage warning low level
9	Aux_Volt_High_Level	Int	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Auxiliary voltage warning High level
10	Clear_Switching_Count	DInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Clear switching Counters
11	RequestFlag	Struct	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	List of the request command to read/Write Value
12	ReadParams	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Command read parameters
13	SetAuxVoltLimits	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Command set voltage level Parameters
14	ClearCycleCount	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Command Clear switching Cycle count
15	System_Command	Struct		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
16	Acknowledge	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Acknowledge Error
17	Store_Switching_Cycles	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	The current Value of the switching Cycle Counters are.
18	Restore_Factory_Setting	Bool	false	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Reset Parameters to default Values

➤ Use siemens function block “IO_LINK_DEVICE” to read and write acyclic data

IO_LINK_DEVICE is a function block wrote by siemens to access acyclic data of an IO-Link module this function block can be used as an alternative to Parker function block “P2M_IOL_BNI005H_PRM” to read or write each single parameter of the P2M/P2H IO-Link slave, the function block library can be download from siemens website.



Parameter	Data Type	Description
REQ	BOOL	Positive edge: Trigger data transfer
ID	HW_IO	Hardware Identifier of IO-Link master module Setting Up P2M/P2H IO-Link slave above
CAP	INT	Access point of the IO-Link communication (Balluff CAP = 255 decimal)
RD_WR	BOOL	Read or write access(0: Read, 1: Write)
PORT	INT	Port number at which the IO-Link device is operated. Add one to port number read from device view (Ex: if the connected port is 0, the port input to function block is equal to 1)
IOL_INDEX	INT	Parameter index (see P2M/P2H IO-Link manual for detail) 2: System Command 64: Switching cycle counter 65: Clear Switching Cycles 66: Channel Error 67: Current Auxiliary Voltage 70: Auxiliary voltage warning levels
IOL_SUBINDEX	INT	Parameter subindex (see P2M/P2H IO-Link manual for detail) 1 to 24: Switching cycle counter, sub index 1 refers to cycle counter of the first output
LEN	INT	Length of the data to be read/written
RECORD_IOL_DATA	ARRAY [0..231] OF BYTE	Source/target area for the data to be read/written